A Level Computer Science

End of Unit Test

Unit 2.2.1.3

Recursion

Name		**Date**	

Score	**Percentage**	**Grade**
/ 24		

Question 1

The layout for a 2-player board game is shown in the table below.

START	1	2	3	4	5	6	7
15	14	13	12	11	10	9	8
16	17	18	19	20	21	22	23
31	30	29	28	27	26	25	24
32	33	34	35	36	37	38	39
47	46	45	44	43	42	41	40
48	49	50	51	52	53	54	55
END	62	61	60	59	58	57	56

The game is played by rolling two 6-sided dice and moving that number of spaces. Both players start on the START space. If a player lands on a space occupied by the other player, they move to the next available space.

The board is to be stored as a 2-dimensional array.

Each time a player moves, a series of obstacles are to be added to the board.

On their turn, each player rolls two dice. The smaller number from the two dice is taken, and that many obstacles will appear on the board in random locations.

For example, if a 3 and 6 are rolled, then 3 obstacles will appear.

A recursive function is written in pseudocode to perform this task.

```
function generateObstacle(diceNumber)
    if diceNumber == 0 then
        return true
    else
        x = randomNumber(0, 7)
        y = randomNumber(0, 7)
        board(x, y) = new obstacle()
        generateObstacle(diceNumber-1)
    endif
endfunction
```

a) The code generates an instance of the object obstacle.

i. Explain the purpose of the code in line `01` in the algorithm.

[2]

ii. Identify the line of code where recursion occurs.

[1]

b) The recursive function could have been written using iteration.

Describe the benefits and drawbacks of using recursion instead of iteration.

i. Benefits

[2]

ii. Drawbacks

[2]

c) Rewrite the function so it uses iteration instead of recursion.

[4]

d) If a position on the board is not occupied, its value is set to a blank string ("").

The current algorithm does not check if the random space generated is currently occupied.

Write a subroutine that takes the generated position of the board, checks if it is free and returns `true` if free, or `false` if occupied.

[3]

Question 2

Many functions can be defined using either recursion or iteration.

a) State one advantage of using recursion instead of iteration.

b) State one disadvantage of using recursion instead of iteration.

[2]

Question 3

Consider the following algorithm in Fig.2, expressed in pseudocode, as a function S:

```
function S(A[0..N-1], value, low, high)

    if (high < low)then
          return error_message
     endif

     mid = (low + high) / 2

    if (A[mid] > value)then
           return S(A, value, low, mid-1)
    elseif (A[mid] < value)then
          return S(A, value, mid+1, high)
    else
          return mid
    endif

endfunction
```

i. Describe what is meant by recursion.

[2]

ii. Identify **one** example of where recursion occurs in this algorithm.

[1]

Question 4

A recursive function, calculate, is shown below:

```
01 function calculate(num1, num2)
02    if num1 == num2 then
03        return num1
04    elseif num1 < num2 then
05        return calculate(num1, (num2-num1))
06    else
07        return calculate(num2, (num1-num2))
08    endif
09 endfunction
```

a) Identify the lines where recursion is used.

[1]

b) Rewrite the function so it uses iteration instead of recursion.

[4]